CASIO

PB-100F

Command List

The BASIC language used on the PB-100F is similar to that used on the PB-100 and PB-300, however a number of commands have been added or altered to improve programming:

Additional Commands: ON-GOTO, ON-GOSUB, LET, REM, BEEP, DATA, READ, RESTORE, PASS

Additional Functions: DEG, DMS$, STR$

Modified Commands:

PB-100F	PB-100 and PB-300
NEW (NEW ALL)	CLEAR (CLEAR A)
CLEAR	VAC
SAVE ALL	SAVE A
LOAD ALL	LOAD A
VERIFY	VER
KEY$	KEY
MID$	MID
DEFM (can be used within a program)	DEFM (can only be used manually)

* Elements can be used repeatedly
{ } Select one of the elements
[ ] Elements can be omitted

Commands

NEW

Function: Program erase. Erases programs and variables.
Format: NEW [ALL]
Examples: NEW
NEW ALL

RUN

Function: Program execution.
Format: RUN [line number]
Examples: RUN
RUN 1000

LIST

Function: Displays the content of a program.
Formats: LIST [ALL]
LIST [starting line number]
Examples: LIST
LIST 50

PASS

Function: Specifies or releases a password.
Format: PASS “character string”
Example: PASS “ABC”

SAVE

Function: Stores a program on a cassette tape.
Formats: SAVE ALL [“file name”]
SAVE [“file name”]
Examples: SAVE
SAVE “MAY”

LOAD

Function: Loads the program from a cassette tape.
Formats: LOAD ALL [“file name”]
LOAD [“file name”]
Examples: LOAD ALL
LOAD “ABC”

VERIFY

Function: Verifies the program stored on a cassette.
Format: VERIFY [“file name”]
Examples: VERIFY
VERIFY “CASIO”

CLEAR

Function: Clears all variables.
Format: CLEAR
Example: CLEAR

END

Function: Terminates program execution.
Format: END
Example: END

STOP

Function: Temporarily suspends program execution.
Format: STOP
Example: STOP

[LET]

Function: Assigns the value of the expression on the right to the variable on the left.
Format: LET {variable=numerical expression}
Example: LET X=12

REM

Function: Statement that expresses a comment.
Format: REM comment
Example: REM SUBROUTINE

INPUT

Function: Inputs data from the keyboard to a variable.
Format: INPUT [“message statement”,] variable name[,[“message statement”,]variable name]*
Example: INPUT A
INPUT “DATA” ,D
INPUT A$, B$, C$

KEY$

Function: A function that enters one character from the keyboard.
Format: KEY$
Example: K$ = KEY$

PRINT

Function: Displays an output element.
Formats: PRINT [output element] [{, or ;} [output element]]*
Examples: PRINT 1/3
PRINT “A=” ; A
PRINT “END” ;

CSR

Function: Displays an output element from a specified location.
Format: CSR numerical expression
Example: PRINT CSRA; “G” ; CSR 9; “H”

GOTO

Function: Unconditionally branches to a specified location.
Formats: GOTO line number
GOTO # program area number
Examples: GOTO 210
GOTO # 4

ON - GOTO

Function: Branches to a specified location according to the branching condition.

Format: ON numerical expression GOTO branching location [, [branching location]]*

Example: ON A GOTO 100 , 200 , 300
ON A GOTO # 3, # 4, # 5, # 7

IF-THEN

Function: When a branching condition is true, the statements after THEN are executed.

Format: IF branching condition THEN statement [: statement]*
IF branching condition THEN branching location

Example: IF K$=”4" THEN N=N+1 : IF N<0 THEN N=0
IF A>B THEN 10
IF C=4 THEN # 3

FOR-TO-STEP-NEXT

Function: Repeats process contained netween FOR and NEXT statements a number of times specified by the control variable.

Format: FOR variable name = numerical expression TO numerical expression [STEP numerical expression] NEXT variable name

Example: FOR I = 0 TO 9: NEXT I

GOSUB

Function: Performs a branching to a specified subroutine.
Formats: GOSUB line number
GOSUB # program area number
Examples: GOSUB 200
GOSUB PROG 3

RETURN

Function: Provides a return from the subroutine to the main program.
Formats: RETURN
Example: RETURN

ON-GOSUB

Function: Branches to a subroutine according to a branching condition.
Formats: ON numerical expression GOSUB [branching location or # program area number][,[branching location or # program area number]]*
Examples: ON A GOSUB 100, 200, 300
ON B GOSUB #2, #3, #4

DATA

Function: Stores data that can be read with a READ statement.
Format: DATA data [, [data]]*
Example: DATA 5, 6, 8, 2

READ

Function: Reads the content of a DATA statement.
Format: READ variable name [, [variable name]]*
Example: READ C, D, E, F

RESTORE

Function: Specifies the location of data to be read by a READ statement.
Format: RESTORE line number
Example: RESTORE 100

PUT

Function: Stores variable data on a cassette tape.
Format: PUT [“file name”] variable 1 [, variable 2]*
Example: PUT “SALES” A, B

GET

Function: Loads data stored on a cassette tape into a variable.
Format: GET [“file name”] variable 1 [, variable 2]*
Example: GET A, B

DEFM

Function: Provides memory expansion.
Format: DEFM numerical expression
Example: DEFM 10

MODE

Function: Sets the state of the computer.
Format: MODE 4 (degrees); MODE 5 (radians); MODE 6 (grades)
MODE 7 (PRT); MODE 8 (releases PRT mode)
Example: MODE 4

SET

Function: Specifies the output format for numerical data.
Format: SET {Fn or En or N}
Example: SET F5:PRINT N

Character Functions

LEN

Function: Gives the length of a character string in a simple character variable.
Format: LEN (simple character variable)
Example: PRINT LEN (B$)

MID$

Function: Fetches a specified number of characters from a specified location of the exclusive character variable ($).
Format: MID$ (location [, number of characters])
Example: PRINT MID$ (X, Y)

VAL

Function: Converts characters in a simple character variable into a numerical value.
Format: VAL (simple character variable)
Example: PRINT VAL (A$)

STR$

Function: Converts the value of a numerical expression into a character string.
Format: STR$ (numerical expression)
Example: PRINT STR$ (45+78)

Numerical Functions

SIN

Function: Trigonometric sine function (sin X)
Format: SIN (numerical expression)
Example: SIN (A/B)

COS

Function: Trigonometric cosine function (cos X)
Format: COS (numerical expression)
Example: COS (A*10)

TAN

Function: Trigonometric tangent function (tan X)
Format: TAN (numerical expression)
Example: TAN (PI/6)

ASN

Function: Inverse trigonometric sine function (arcsin or sin-1)
Format: ASN (numerical expression)
Example: ASN (X*X)

ACS

Function: Inverse trigonometric cosine function (arccos or cos-1)
Format: ACS (numerical expression)
Example: ACS (A+12)

ATN

Function: Inverse trigonometric tangent function (arctan or tan-1)
Format: ATN (numerical expression)
Example: ATN (A/100)

LOG

Function: Common logarithmic function
Format: LOG (numerical expression)
Example: LOG (2.71828)

LN

Function: Natural logarithmic function
Format: LN (numerical expression)
Example: LN (1.6754)

EXP

Function: Exponential function
Format: EXP (numerical expression)
Example: EXP (1)

SQR

Function: Square root
Format: SQR (numerical expression)
Example: SQR (30)

ABS

Function: Gives the absolute value of the numerical expression.
Format: ABS (numerical expression)
Example: ABS (-10.5)

SGN

Function: Gives the sign of the numerical expression.
Format: SGN (numerical expression)
Example: SGN (-1)

FRAC

Function: Gives the value of the fractional part of the numerical expression.
Format: FRAC (numerical expression)
Example: FRAC (2.64)

RND

Function: Gives the value obtained by rounding the specified digit.
Format: ROUND (numerical expression, digit position)
Example: ROUND (1.414, 2)

INT

Function: Gives the largest integer which is less than or equal to the specified numerical expression.
Format: INT (numerical expression)
Example: INT (3.14)

RAN#

Function: Gives a random number from 0 to 1.
Format: RAN#
Example: INT (RAN# * 10)

DEG

Function: Converts sexagesimal to decimal.
Format: DEG (degree, minute, second)
Example: DEG (12, 34, 56)
PRINT DEG (A, B, C)

DMS$

Function: Converts decimal to sexagesimal.
Format: DMS$(numerical expression)
Example: DMS$(45.678)
$=DMS$(A)

Error Message Table

Error	Meaning	Cause
1	Memory overflow or operator level overflow	- Number of steps are insufficient. Program cannot be written. - Operator level overflow.
2	Syntax error	- Format error in program, etc. - Left-hand and right-hand formats differ in an assignment statement etc.
3	Mathematical error	- The result of a numerical expression calculation exceeds +- 1 x 10^100. - The argument of numerical function is outside the input range. - Result is indefinite or impossible.
4	Un-definition error	- No designated line number for GOTO or GOSUB statement. - A READ or READ# statement is executed when there is no data to be read.
5	Argument error	- For a command or function that requires an argument, the argument is outside the input range.
6	Variable error	- Attempt was made to use memory which has not been expanded. - Attempt was made to use the same memory for a numerical variable and a character variable at the same time.
7	Nesting error	- RETURN statement is executed when subroutine is not being executed. - NEXT statement is executed when not in FOR loop. - Subroutine nesting levels exceed 8. - FOR-NEXT loop nesting levels exceed 4.
8	Password	- When the password is specified, (a) another password is specidied. (b) a command such as LIST or NEW which cannot be used is executed.
9	Option error	- SAVE or PUT command was executed without a tape recorder connected. - Input signal used by LOAD or GET command cannot be read. - Printer battery is weak or printer paper is jammed.

Specifications

- Calculation Range

+-1 x 10^99 to +_9.999999999 x 10^99 and 0 (internal calculations use 12-digit mantissa)

- Number of steps

Maximum 544 steps (maximum 1,568 steps when optional RAM pack is loaded)

- Program capacity

Maximum 10 programs (P0 through P9)

- Number of variables

Standard 26, expandable to 94 (maximum 222 variables when optional RAM pack is loaded) and exclusive character variable ($)

- Nesting

Subroutine - 8 levels
FOR-NEXT loop - 4 levels
Numerical value - 6 levels
Operators - 12 levels

- Display system and contents

10-digit mantissa (including minus sign) or 8-digit mantissa (7 digits for negative number) and 2-digit exponent.

- Power supply

2 lithium batteries (CR2032)

- Power consumption

Maximum 0.02W

- Battery life (Continuous use)

Mainframe only - approximately 170 hours
With options connected - approximately 100 hours

- Auto power-off

Power is turned off automatically approximately 7 minutes after last operation.

Character Code Table

	SPACE	+	-	*	/	↑	!	″	#	$	>	≥	=	≤	<	≠
Numbers	0	1	2	3	4	5	6	7	8	9	.	π	)	(	Ẽ	E
Capital letters	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	Q	R	S	T	U	V	W	X	Y	Z						
Small letters	a	b	c	d	e	f	g	h	I	j	k	l	m	n	o	p
	q	r	s	t	u	v	w	x	y	z						
Symbols	?	,	;	:												
Graphic symbols	◯	∑	o	Δ	@	X	÷	♠	←	♥	♦	♣	μ	Ω	↓	→
	%	¥	□	[	&	_	‘	·	]	■						

CASIO

Digital Reconstruction: Steven Fox 2021